

# Syntax-Ignorant N-gram Embeddings for Sentiment Analysis of Arabic Dialects

Hala Mulki<sup>\*§</sup>, Hatem Haddad<sup>†§</sup>, Mourad Gridach<sup>\*\*</sup> and Ismail Babaoğlu<sup>\*</sup>

<sup>\*</sup>Department of Computer Engineering, Konya Technical University, Turkey

<sup>†</sup>RIADI Laboratory, National School of Computer Sciences, University of Manouba, Tunisia

<sup>\*\*</sup>Computational Bioscience Program, University of Colorado, School of Medicine, USA

<sup>§</sup>iCompass Consulting, Tunisia

halamulki@selcuk.edu.tr, haddad.Hatem@gmail.com

mourad.gridach@ucdenver.edu, ibabaoğlu@selcuk.edu.tr

## Abstract

Arabic sentiment analysis models have employed compositional embedding features to represent the Arabic dialectal content. These embeddings are usually composed via ordered, syntax-aware composition functions and learned within deep neural frameworks. With the free word order and the varying syntax nature across the different Arabic dialects, a sentiment analysis system developed for one dialect might not be efficient for the others. Here we present syntax-ignorant n-gram embeddings to be used in sentiment analysis of several Arabic dialects. The proposed embeddings were composed and learned using an unordered composition function and a shallow neural model. Five datasets of different dialects were used to evaluate the produced embeddings in the sentiment analysis task. The obtained results revealed that, our syntax-ignorant embeddings could outperform word2vec model and doc2vec both variant models in addition to hand-crafted system baselines, while a competent performance was noticed towards baseline systems that adopted more complicated neural architectures.

## 1 Introduction

According to the used features, existing Arabic Sentiment Analysis (ASA) systems can be classified into: (a) hand-crafted-based systems (Abdulla et al., 2013; El-Beltagy et al., 2017) where linguistic/stylistic and lexical features are generated by morphological analyzers and semantic resources and (b) text embeddings-based systems that adopt word/sentence embeddings using one of the composition models (Gridach et al., 2017; Medhaffar et al., 2017). While the first type of ASA systems provide a comparable performance, the generation of hand-crafted features is considered a labor-intensive task that requires using language/dialect-specific NLP tools and techniques (Altowayan and

Tao, 2016). In contrast, text embeddings-based systems can use the raw unprocessed input content to generate expressive features to represent words or even longer pieces of text through using the composition models (Mikolov et al., 2013).

Composition models aim to construct a phrase/sentence embeddings based on its constituent word embeddings and structural information (Iyyer et al., 2015). Two main types of these models can be recognized: (a) Ordered models where the order and linguistic/grammatical structure of the input words do count while constructing the phrase/sentence vector and (b) Unordered models in which the word representations are combined irrespective of their order using algebraic operations (Sum of Word Embeddings (SOWE), average (Avg), mean and multiplication functions) (Mitchell and Lapata, 2010).

Context words along side their syntactic properties have been considered essential to build effective word embeddings able to infer the semantic/syntactic similarities among words, phrases or sentences. Consequently, most of the recently-developed SA systems adopted deep neural network architectures such as Convolutional Neural Networks (CNNs) and Recursive Neural Networks (RecNNs) where ordered composition models are employed to grasp the syntactic and linguistic relations between the words (Al Sallab et al., 2015; Dahou et al., 2016). These systems required more training time to learn words' order-aware embeddings due to the high computational complexity consumed at each layer of the model (Iyyer et al., 2015). However, such embeddings resulting from ordered compositionality might not form discriminating features for the Arabic dialects; especially that these dialects have a free word order and varying syntactic/grammatical rules (Brustad, 2000). For instance, the dialectal (Levantine) sentence in-

هاالفكرة	انا	حببنا
O	S	V
هاالفكرة	حببنا	انا
O	V	S
هاالفكرة	انا	حببنا
V	S	O
انا	حببنا	هاالفكرة
S	V	O

Table 1: Free word order of dialectal Arabic.

Dialect	Sentence	POS
Levantine	الوضع ماشي الحال The situation is <b>okay</b>	Adjective
Moroccan	نحن ماشي سعداء We are <b>not</b> happy	Negation
Egyptian	كنت ماشي فاتجاه البيت I was <b>walking</b> towards home	Verb

Table 2: Syntactic differences across the Arabic dialects.

investigated in Table 1 meaning “I liked this idea” can be represented by several word orders: VSO, SVO, OSV and OVS and yet, implies the same meaning and sentiment.

On the other hand, the Arabic dialects show phonological, morphological, lexical, and syntactic differences such that the same word might infer different syntactic information across different dialects. To clarify that, Table 2 reviews how the word “ماشي” has several Part Of Speech (POS) tags, multiple meanings and different sentiments across three Arabic dialects.

Thus, to handle such informality of DA, we propose an unordered composition model to construct sentence/phrase embeddings regardless of the order and the syntax of the context’s words. Nevertheless, when coming to the sentiment analysis task, sentence embeddings that are merely composed and learned based on the context words do not always infer the sentiment accurately. This is due to the fact that, some words of contradict sentiments might be mentioned within identical contexts which leads to map opposite words close to each other in the embedding space. To clarify that, both sentences in Example 1 and Example 2 contain the same context words organized in the same order; yet the first sentence is of positive polarity while the second has a negative sentiment since the words “ممتع” and “ممل” are antonyms that

mean “interesting” and “boring”, respectively.

**Example 1** هالفيلم ممتع بشكل ما بينوصف<sup>1</sup>

**Example 2** هالفيلم ممل بشكل ما بينوصف<sup>2</sup>

One way to address this issue is to learn the embeddings from sentiment-annotated corpora such that the sentiment information is incorporated along with the contextual data within the composed embedding during the training phase. This was examined with the English language, as Tang et al. (2014) presented sentiment-specific word embeddings (SSWE) composed via unordered Min, Max and Avg composition models. Another pairing between Avg composition functions and supervised learning was introduced by (Iyyer et al., 2015) where a neural model of two hidden layers called Deep Averaging Neural network (DAN) was used to learn the embeddings together with sentiment, yielding a performance competent to much more complicated models such as RecNNs and CNNs-Multi Channel (CNN-MC).

While some of the recent ASA systems considered the syntactic information in the composed embeddings (Al Sallab et al., 2015), other models used pretrained or unsupervised unordered word/doc embeddings as features to mine the sentiment of MSA/DA content (Altowayan and Tao, 2016; Gridach et al., 2017). However, mining the sentiment of DA using syntax-aware ordered embeddings might be ineffective especially with the drastic differences between Eastern and Western Arabic dialects (Brustad, 2000). In addition, for the SA task, the embeddings learned from unlabeled data are not as discriminating as those learned with sentiment information integrated in the embedding vectors (Tang et al., 2014). This evokes the need to provide a sentiment-specific, dialect-independent embeddings with which the gap resulted from the differences among Arabic dialects can be bridged. Such embeddings would ignore the syntactic structure and focus on the semantic and sentiment information.

Inspired by (Iyyer et al., 2015; Tang et al., 2014), we hypothesize that representing a sentence by its constituent sentiment-specific, unordered and syntax-ignorant n-gram embeddings can handle the diversity of the Arabic dialects and provide better features for the dialectal Arabic SA task. In the current paper, we present a SA

<sup>1</sup>This movie is incredibly interesting.

<sup>2</sup>This movie is incredibly boring.

framework whose features are n-gram embeddings learned from labeled data (sentiment-specific) and composed via the additive unordered composition function (syntax-ignorant) known as SOWE. The embeddings composition and the sentiment learning processes were conducted within Tw-StAR framework which forms a shallow feed-forward neural network of single hidden layer. The contributions of this study can be briefly described as follows:

1. Based on the outperformance of SOWE composition function in sentence semantic similarity applications (White et al., 2015), we believe that SOWE can be an effective replacement of the Average (Avg) composition functions used in (Iyyer et al., 2015) and (Mikolov et al., 2013). Besides its low computation complexity as it conducts an element-wise sum over the word embedding vectors contained in a sentence, SOWE can capture and encode semantic and synonymous information in the resulting composed embeddings (White et al., 2015).
2. Given that, DA has a free word order and a varying syntactic nature, therefore, unlike (Tang et al., 2014) whose embeddings were generated using corrupted input n-grams from which the syntactic context nature are learned, we feed whole n-grams to our model as the training objective is to capture the semantic and sentiment relations regardless of the order and the syntax of the context words.
3. In contrast to previous studies, that composed unordered embeddings within deep neural models (Iyyer et al., 2015), the embeddings introduced here are generated and learned within a shallow feed-forward neural model as we are seeking to investigate whether SA of DA can be performed using less complicated neural architectures.

## 2 The Proposed Model (Tw-StAR)

As we are seeking to answer the question: To which extent a shallow neural model, trained with embeddings specifically formulated to target DA, can rival complicated neural architectures?, we chose to implement Tw-StAR as a feed-forward neural network in which sentiment-

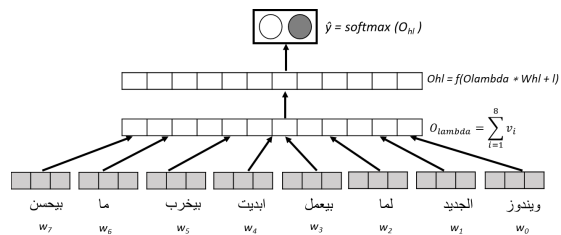


Figure 1: Tw-StAR neural sentiment analysis model.

specific, syntax-ignorant and semantic-enriched n-grams embeddings are composed using SOWE function and learned in a supervised manner. The generated n-gram embeddings were then employed as discriminative features to predict the positive/negative sentiment of the tackled input sentences. As it is shown in Figure 1, Tw-StAR model is a shallow feed-forward neural network composed of the following layers: the input or embeddings layer followed by lambda layer then a hidden layer and finally an output layer with softmax function applied for the classification into positive or negative sentiment.

### 2.1 Model Description

The embedding layer, in Tw-StAR, acts as a word lookup table, it is responsible of projecting words in the input into their corresponding dense vector representations. Given the input sentences, in order to handle their varying lengths, each sentence  $S$  of  $l$  words was formulated as a sequence of fixed-length n-grams generated using a sliding window of a specific size  $C$ . Instead of using corrupted input n-grams as in the SSWEu model provided in (Tang et al., 2014) and CBOW in (Mikolov et al., 2013), whole n-grams were fed to the embedding layer such that each n-gram is accompanied with the sentiment label of the sentence from which it was derived; where  $[1,0]$  and  $[0,1]$  vectors were used to represent the positive and negative polarities, respectively. Having the n-grams prepared, their constituent words are mapped into the corresponding embeddings using the weights matrix  $M \in \mathbb{R}^{|V| \times d}$  of the embedding layer, where  $|V|$  is the vocabulary size and  $d$  denotes the embedding dimension.

The weights of the embedding layer were initialized randomly using Glorot uniform initializer (Glorot and Bengio, 2010) then optimized while training the model. It should be noted that, we chose not to use pretrained word embeddings for

initialization, as the available Arabic pretrained word embeddings from (Zahran et al., 2015) and (Al-Rfou et al., 2013) were generated based on MSA/Egyptian corpora. We assume that, this can lead to out-of-vocabulary (OOV) issues especially with the Tunisian and Moroccan content, used in this study, where less common words with MSA/Egyptian do exist. Thus, for a single fixed-length n-gram containing a sequence of words  $\{w_i, w_{i+1}, w_{i+2}, \dots, w_{i+C-1}\}$ , each word  $w_i$  is represented by a unique integer index  $i \in [0, V]$  and stored as a one-hot vector  $vec_i$  whose values are zero in all positions except at the  $i$ -th index. To obtain the embedding vector  $v_i$  of a word  $w_i$ , its one-hot vector  $vec_i$  is multiplied by the matrix  $M$  as in equation (1).

$$v_i = vec_i * M \in R^{1 \times d} \quad (1)$$

As each row of the embedding matrix  $M$  denotes the dense embedding representation of a specific word in the vocabulary, multiplying the one-hot vector of each word in the input by the embedding matrix  $M$ , will essentially select one of  $M$  rows that corresponds to the embeddings of this word.

The resulting word embeddings were then combined using the compositional model SOWE which is applied by the next linear layer Lambda. In this layer, an element-wise sum is conducted over the word embedding vectors. Here we could refer to the fact that, although the n-gram scheme retains the local order of its constituent words, formulating the n-gram embeddings vector via the additive function SOWE, totally ignores the words' order since an identical embedding vector would be composed for any order of the words contained in an n-gram. Thus, the output of the lambda layer is a single embeddings vector  $O_{lambda} \in R^{1 \times |d|}$  resulted from summing the embeddings vectors produced by the embedding layer which correspond to the input words contained in a window of size  $C$ :

$$O_{lambda} = \sum_{i=1}^C v_i \in R^{1 \times d} \quad (2)$$

In the subsequent hidden layer ( $hl$ ), the output from the previous layer  $O_{lambda}$  is subjected to a linear transformation using the weights matrix  $W_{hl} \in R^{d \times 2}$  and biases  $b_{hl} \in R^{1 \times 2}$ :

$$O_{hl} = f(O_{lambda} * W_{hl} + b_{hl}) \in R^{1 \times 2} \quad (3)$$

Where  $W_{hl}$  and  $b_{hl}$  form the model's parameters that are learned and optimized during the training process and  $f$  refers to the activation function that introduces non-linear discriminative features to our model. Here, we used Hard sigmoid activation function ( $h_\sigma$ ). Hard sigmoid is a piecewise function whose output are very similar to the traditional sigmoid, however, it is computationally cheaper which leads to a smarter model since it accelerates the learning process in each iteration (Gulcehre et al., 2016).

Finally, the output  $O_{hl}$  resulting from the hidden layer is forwarded into the output layer ( $Ol$ ) where a softmax function is applied to induce the estimated probabilities for each output label (positive/negative) of a specific n-gram. Where each n-gram is accompanied with the predicted two dimensional label  $[1,0]$  denoting positive or  $[0,1]$  indicating negative.

$$\hat{y} = softmax(O_{hl}) \in R^{1 \times 2} \quad (6)$$

Softmax selects the maximum score among the two predicted conditional probabilities to denote positive or negative polarity of an input n-gram where the distribution of the form  $[1,0]$  was assigned for positive while  $[0,1]$  distribution form was adopted for negative. Thus, if the gold sentiment polarity of an n-gram is positive, the predicted positive score should be higher than the negative score while if the gold sentiment polarity of a word sequence is negative, its positive score should be smaller than the negative score. To decide the polarity of the whole sentence, the predicted positive scores and negative scores of n-grams are summed then each of which is divided by the number of the n-grams contained in this sentence resulting two values representing the potential positive and negative scores of the input sentence. The final sentence polarity is, thus, decided according to the greater among these two values. Cross-entropy loss between gold sentiment distribution and predicted distribution was adopted such that the loss function of the model:

$$J(\theta) = - \sum_{k=\{0,1\}} y_k \log \hat{y}_k \quad (7)$$

Where  $y \in R^2$  is the gold sentiment value represented by a one-hot vector,  $\hat{y}$  is the sentiment distribution predicted by the model while  $\theta$  refers to the parameters (weights and biases) of the model to be learned and optimized during the training process.

Dataset	Train	Dev	Test	Voc.
ArTwitter	1,280	320	400	7,253
TEC	1,948	487	608	10,675
TSAC	4,680	1,170	1,516	17,741
MEC	6,561	1,641	2,051	37,888
MDT	2,747	687	860	16,450

Table 3: The statistics of the used datasets.

## 2.2 Training details and Model’s Parameters

The key hyper parameters of the proposed model are the sliding window size  $C$  and the embeddings dimension  $d$ . We have selected both parameters’ values empirically during the model tuning period.

To train the proposed neural network, the back-propagation algorithm with Adaptive Moment estimation (Adam) stochastic optimization method (Kingma and Ba, 2014) has been used. Adam optimizer combines the early optimization speed of Adagrad with the better later convergence of various other methods like Adadelta and RMSprop. This is done through calculating learning rates and storing momentum changes for each model parameter separately.

To deal with the overfitting issue, Dropout was used as a regularization mechanism. The value of the dropout parameter was selected empirically during the model’s tuning period.

## 3 Experimental Study

### 3.1 Datasets

For the model evaluation, Tw-StAR was employed to predict the sentiment in five publicly available datasets (See Table 3). Four of them were written in Eastern (Jordanian) and Western (Tunisian, Moroccan) Arabic dialects, while the fifth combined Eastern, Western and Gulf Arabic dialects. They are as follows:

- Arabic Twitter Dataset (ArTwitter): combines 2,000 positive/negative tweets mostly written in the Jordanian dialect (Abdulla et al., 2013).
- Tunisian Election Corpus (TEC): refers to 3,043 tweets positive/negative combining MSA and Tunisian dialect where Tunisian tweets form the majority of the data (Sayadi et al., 2016).
- Tunisian Sentiment Analysis Corpus (TSAC): combines 7,366 positive/negative Facebook comments (Medhaffar et al., 2017).

Data	C=6	C=7	C=8	C=9	C=10
ArTwitter	82.7	83.0	<b>83.3</b>	82.3	81.5
TEC	87.6	<b>87.9</b>	<b>87.9</b>	83.6	81.2
TSAC	86.1	85.9	<b>86.6</b>	86.5	86.3
MEC	63.9	<b>68.6</b>	<b>68.6</b>	67.1	66.5
MDT	73.4	73.4	<b>73.8</b>	73.3	72.5

Table 4: F-measure values (%) obtained with dev sets for different window sizes.

- Moroccan Election Corpus (MEC): combines 10,253 positive/negative Facebook comments (Elouardighi et al., 2017).
- Mixed-Dialects Tweets (MDT) (Altowayan and Tao, 2016): forms a combination of 4,294 positive/negative tweets from three datasets of MSA and dialectal content including: (a) Jordanian: Artwitter (Abdulla et al., 2013), (b) Egyptian: ASTD (Nabil et al., 2015) and (c) Multiple dialects: QCRI (Mourad and Darwish, 2013).

### 3.2 Results and Discussion

The model’s parameters ( $C$ ,  $d$ , dropout) were assigned empirically. Among several window sizes ranging from 6 to 10, a window size value equals to 8 was adopted since it produced the best F-measure in all datasets as it is shown in Table 4. Consequently, each input sentence is represented by a set of 8-grams to be fed to the model. Similarly, upon examining three embedding dimensions values equal to 50, 100 and 150, and several dropout values ranging from 0.2 to 0.5,  $d=100$  and dropout=0.2 were adopted for dimensions and dropout, respectively.

The efficiency of the proposed n-gram embeddings composed by SOWE were compared against word embeddings (word2vec) and document embeddings (doc2vec). Using a supervised learning strategy with sentiment labels included in the training corpora, and provided with the same parameters of Tw-StAR model in terms of window size and embedding dimensions, we trained word2vec (Mikolov et al., 2013) and doc2vec (Pv-DBow/Pv-DM) (Le and Mikolov, 2014) algorithms on each of the tackled datasets to generate the proper embedding features. In the distributed bag of words (DBow), the embeddings vector representing a sentence is composed with words’ order ignored, whereas the distributed memory variant (DM) follows the CBOW mechanism as it considers the words order while learning the

Dataset	Model	P. (%)	R. (%)	F1 (%)	A. (%)
ArTwitter	Combined LSTMs (Al-Azani and El-Alfy, 2017)	<b>87.3</b>	<b>87.3</b>	<b>87.2</b>	<b>87.2</b>
	CNNs (Dahou et al., 2016)	-	-	-	85.0
	word2vec	72.0	71.9	71.9	72.0
	doc2vec (DM)	61.2	60.7	60.1	60.4
	doc2vec (DBoW)	63.1	60.6	58.2	59.9
	<b>Tw-StAR</b>	<b>85.4</b>	<b>84.9</b>	<b>84.8</b>	<b>84.9</b>
TEC	hand-crafted (Sayadi et al., 2016)	67.0	71.0	63.0	71.1
	word2vec	62.6	59.7	58.4	61.9
	doc2vec (DM)	65.6	59.3	56.4	62.2
	doc2vec (DBoW)	62.9	58.9	56.7	61.4
	<b>Tw-StAR</b>	<b>87.4</b>	<b>88.4</b>	<b>87.8</b>	<b>88.2</b>
TSAC	MLP (Medhaffar et al., 2017)	78.0	78.0	78.0	78.0
	word2vec	78.0	77.2	77.4	78.2
	doc2vec (DM)	61.0	58.3	57.2	61.7
	doc2vec (DBoW)	55.9	54.1	52.1	58.0
	<b>Tw-StAR</b>	<b>86.2</b>	<b>86.3</b>	<b>86.2</b>	<b>86.5</b>
MEC	hand-crafted (Elouardighi et al., 2017)	-	-	-	78.0
	word2vec	63.6	64.0	63.8	69.1
	doc2vec (DM)	74.7	65.0	66.4	76.6
	doc2vec (DBoW)	60.4	56.6	56.4	69.3
	<b>Tw-StAR</b>	<b>76.2</b>	<b>71.2</b>	<b>72.8</b>	<b>79.2</b>
MDT	Arabic word embeddings (Altowayan and Tao, 2016)	<b>83.0</b>	<b>76.5</b>	<b>79.6</b>	<b>80.2</b>
	word2vec	59.3	59.2	59.2	59.4
	doc2vec (DM)	58.5	57.9	57.4	58.4
	doc2vec (DBoW)	61.2	59.4	58.2	60.2
	<b>Tw-StAR</b>	<b>75.8</b>	<b>74.3</b>	<b>74.3</b>	<b>74.8</b>
Average	word2vec	67.1**	66.4*	66.1*	68.1**
	doc2vec (DM)	64.2*	60.2**	59.5**	63.8*
	doc2vec (DBoW)	60.1**	57.9**	56.3**	61.7**
	<b>Tw-StAR</b>	<b>82.2</b>	<b>81.0</b>	<b>81.2</b>	<b>82.7</b>

Table 5: Tw-StAR performances against baseline systems and word2vec/doc2vec for all datasets. (\*, \*\*, \*\*\*) refers to a significant difference at P-value < 0.05, < 0.01, < 0.001, respectively, compared to Tw-StAR.

composed sentence embeddings vector (Le and Mikolov, 2014). Having the word embeddings and document embeddings generated for each dataset by word2vec and doc2vec algorithms, they were used as features to train Tw-StAR neural model on recognizing the sentiment of the datasets in Table 3. This was done through replacing the embeddings layer in Tw-StAR by the embeddings produced by word2vec and both variants of doc2vec. It should be noted that, word2vec and both variants of doc2vec were trained in a supervised manner. Thus, their learned embeddings are sentiment informed as the polarity labels were associated with the input training instances. This enabled a fair comparison between word2vec/doc2vec variants and our sentiment-specific syntax-ignorant n-grams embeddings.

Table 5, reviews the sentiment classification performances achieved using n-grams by SOWE, word vectors by word2vec and sentence vectors by doc2vec (PV-DBoW/PV-DM) for all datasets. The obtained performances of Tw-StAR were further compared against the baseline systems that tack-

led the same datasets and also listed in Table 5; where P., R., F1 and A. denote the achieved averaged precision, recall, F-measure and accuracy respectively. It should be mentioned that, due to the limited work in SA of under-represented dialects such as Tunisian and Moroccan, it wasn't possible to perform the comparison against text embeddings-based baselines for these dialects, as the provided models for MEC and TEC datasets used only hand-crafted features.

The results in Table 5 suggest the outperformance of the proposed embeddings over those generated by word2vec and doc2vec for most datasets. This was emphasized through the significance test (T-test), where the sentiment classification performance of Tw-StAR with n-grams embeddings used for training was proved to be significantly better than that produced with word2vec/doc2vec embedding features. For instance, the best achieved F-measure was in TEC dataset with a value of 87.7% compared to 58.4%, 56.4% and 56.7% scored by word2vec, doc2vec (PV-DM) and doc2vec (PV-DBoW), respectively.

This could be explained by the ability of SOWE to capture the semantic information along with the synonymous relations among words more accurately than the average function used by doc2vec variants (White et al., 2015). On the other hand, it can be seen from Table 5 that, for datasets having an MSA-dominated content such as MEC, doc2vec (PV-DM) performs better than word2vec and doc2vec (PV-DBoW). Indeed, the achieved accuracy for MEC dataset with the embeddings learned by doc2vec (PV-DM) was 76.6% compared to 69.1% and 69.3% scored by word2vec and doc2vec (PV-DBoW), respectively. This could be due to the fact that, doc2vec (PV-DM) is a syntax-aware embeddings learning method where it acts as a memory that remembers what is missing from the context to predict a (typically) center word (Le and Mikolov, 2014). Therefore, it can handle the MSA-dominated data where syntax does matter in indicating the sentiment.

Compared to the state-of-the-art applied on the tackled datasets, our results showed that Tw-StAR trained with the proposed embeddings could improve the performance over the baselines in most of the datasets. As we can see in Table 5, with Tw-StAR applied, the accuracy increased by 17.1%, 8.3% and 1.2% for TEC, TSAC and MEC datasets, respectively. On the other hand, the less accuracy increment was reported in MSA/Moroccan MEC dataset; This defines the proposed embeddings as expressive features of pure dialectal content more than they are of MSA. Since the free word order and varying syntactic structure of dialects can be better handled by SOWE. Moreover, for ArTwitter dataset, a competent performance was achieved by Tw-StAR against complicated neural architectures such as CNNs adopted by (Dahou et al., 2016) and combined LSTMs used in (Al-Azani and El-Alfy, 2017), where the accuracy decreased by 0.1% and 2.3% compared to (Dahou et al., 2016) and (Al-Azani and El-Alfy, 2017), respectively. Hence, a shallow neural model such as Tw-StAR trained with embeddings specifically composed to target the DA content can rival much more complicated neural architectures. In addition, for MDT dataset that contains three different dialects, although Tw-StAR could not outperform the baseline system, a satisfying performance was achieved without the need for a huge training corpus used by (Altowayan and Tao, 2016).

Aiming to inspect the performance of the n-




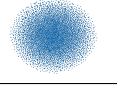
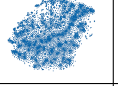
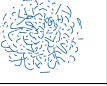


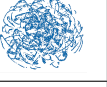



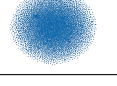


Dataset	word2vec	doc2vec	Tw-StAR
ArTwitter			
TEC			
TSAC			
MEC			
MDT			

Figure 2:  $t$ -SNE visualization of word vectors learned by word2vec/doc2vec against word vectors learned by Tw-StAR.

gram embeddings more deeply, we visualized the embedding vectors learned by Tw-StAR against word vectors generated by word2vec and paragraph vectors learned via doc2vec (PV-DBoW). This is done by projecting the embedding vectors into a two dimensional space using the  $t$ -Distributed Stochastic Neighbour Embedding ( $t$ -SNE) technique (Maaten and Hinton, 2008).

Considering Figure 2, a clustering behavior of the words that compose  $n$ -grams or document embeddings could be observed in both doc2vec (PV-DBoW) and Tw-StAR models. In word2vec model, however, word vectors tend to spread sparsely in the embeddings space. This was reflected on the performance of the embeddings as discriminating features for the SA task. To clarify that, considering TSAC dataset, we have noticed that pure Tunisian dialectal words such “إنحبوك” and “باهي”<sup>3</sup> which bear positive sentiments were mapped by Tw-StAR model close to each other in the embeddings space. However, when looking to the representations created for the same dataset by doc2vec (PV-DBoW), we have come through the words “إنحبوك” and “هايلة”<sup>4</sup> which refer to a positive sentiment, yet they are mapped close to the negative words “ممسطها” and “خامج”<sup>5</sup> in the embeddings space.

<sup>3</sup>We love you and good.

<sup>4</sup>We love you and excellent.

<sup>5</sup>Dull and a dirty man.

## 4 Related works

In (Altowayan and Tao, 2016), Arabic word vectors were generated through training Continuous Bag of Words (CBOW) algorithm (Mikolov et al., 2013) using an Arabic corpus of 190 million words. To evaluate the generated embeddings, they were used to train several binary classifiers on recognition of the subjectivity and sentiment polarity in a combination of twitter datasets: ASTD (Nabil et al., 2015), ArTwitter (Abdulla et al., 2013) and QCRI (Mourad and Darwish, 2013) and MSA news articles. The model's performance was slightly better than (Mourad and Darwish, 2013) in subjectivity classification, while for the polarity classification of the twitter datasets, the best metric values were scored by the Nu-SVM with an accuracy of 80.21% and an F-measure of 79.62%.

A study by (Dahou et al., 2016) introduced a CNN-based deep learning SA model. The model was trained with word embeddings learned from a corpus of 3.4 billion Arabic words using CBOW and Skip-Gram (SG). Using CNN as a building unit, a neural model with one non-static channel and one convolutional layer was developed. Multiple filter window sizes were adopted to perform the convolutional operation while a max-over-time pooling layer was utilized to capture the most relevant global features (Collobert et al., 2011). The model was applied on several datasets such as ASTD (Nabil et al., 2015), ArTwitter (Abdulla et al., 2013). The results revealed that the performance of the presented model mostly outperformed all the state-of-the-art systems where for ArTwitter, the achieved accuracy was 85.0%.

The idea of including Arabic pre-trained word embeddings in a deep neural SA model was introduced by (Gridach et al., 2017). The authors used word embeddings provided by (Zahran et al., 2015) previously trained with MSA/dialectal corpora by Glove, SG and CBOW methods. These embeddings were used to initialize the input word embeddings with which their model CNN-ASAWR was trained. The proposed model was developed as a variant of (Collobert et al., 2011) system and customized to conduct SA on two MSA/dialectal datasets: ASTD (Nabil et al., 2015) and SemEval-2017 (El-Beltagy et al., 2017). Results showed that using pre-trained word embeddings led to better evaluation measures compared to the baseline systems. In ASTD dataset for instance, the best F-measure

scored by CNN-ASAWR was 72.14% compared to 62.60% achieved by (Nabil et al., 2015) while for SemEval-2017, an F-measure of 63% was achieved against 61% scored by the system of (El-Beltagy et al., 2017).

As a first attempt to leverage document embeddings in ASA, doc2vec model was used in (Medhaffar et al., 2017) to generate training vectors for a Tunisian SA model. The presented model was evaluated using a combination of publicly available MSA/multi-dialectal datasets and a manually annotated Tunisian Sentiment Analysis Corpus (TSAC) obtained from Facebook comments about popular TV shows. The input data was represented by document vectors which were used later to train SVM, Bernoulli NB (BNB) and Multilayer Perceptron (MLP) classifiers. The best results were scored by a multi-layer perceptron (MLP) classifier when TSAC corpus was solely used as a training set where it achieved an accuracy equals to 78% and an F-measure value of 78%.

## 5 Conclusion

We introduced syntax-ignorant, n-gram embeddings as discriminating features in the context of sentiment analysis of Arabic dialects. The presented model Tw-StAR trained with these embeddings could classify the sentiment of several dialects better than most baseline systems. Being composed via SOWE function, our embeddings emphasized the efficiency of using unordered additive composition model in SA as the produced performances by n-gram embeddings were better than those learned via word2vec and doc2vec (PV-DM/PV-DBoW) models. Based on the visualization of the word embeddings learned by Tw-StAR, word2vec and doc2vec (PV-DBoW) models, it was possible to deduce that several words of close sentiments were better mapped using Tw-StAR model. Finally, it was revealed that, for Arabic dialects, a shallow neural model trained with unordered embeddings can address the varying syntactic structure and free word order issues yielding a competent performance with much more complicated deep learning architectures. A natural future step would involve using the proposed embeddings to represent the sentiment of other languages. Furthermore, a multi-dialectal lexicon would be constructed based on the distances among the word embedding vectors learned via Tw-StAR and visualized by *t*-SNE tool.



## References

- Nawaf A. Abdulla, Nizar A. Ahmed, Mohammed A. Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6. IEEE.
- Sadam Al-Azani and El-Sayed M. El-Alfy. 2017. Hybrid deep learning for sentiment polarity determination of arabic microblogs. In *International Conference on Neural Information Processing*, pages 491–500. Springer.
- Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*.
- Ahmad Al Sallab, Hazem Hajj, Gilbert Badaro, Ramy Baly, Wassim El Hajj, and Khaled Bashir Shaban. 2015. Deep learning models for sentiment analysis in arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 9–17.
- Aziz A. Altowayan and Lixin Tao. 2016. Word embeddings for arabic sentiment analysis. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3820–3825. IEEE.
- Kristen E. Brustad. 2000. *The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects*. ERIC.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.
- Abdelghani Dahou, Shengwu Xiong, Junwei Zhou, Mohamed Houcine Haddoud, and Pengfei Duan. 2016. Word embeddings and convolutional neural network for arabic sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2418–2427.
- Samhaa R. El-Beltagy, Mona El kalamawy, and Abu Bakr Soliman. 2017. Niletmrg at semeval-2017 task 4: Arabic sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 790–795. Association for Computational Linguistics.
- Abdeljalil Elouardighi, Mohcine Maghfour, Hafdalla Hammia, and Fatima-zahra Aazi. 2017. A machine learning approach for sentiment analysis in the standard or dialectal arabic facebook comments. In *3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pages 1–8. IEEE.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.
- Mourad Gridach, Hatem Haddad, and Hala Mulki. 2017. Empirical evaluation of word representations on arabic sentiment analysis. In *International Conference on Arabic Language Processing (ICALP)*, pages 147–158. Springer.
- Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016. Noisy activation functions. In *International Conference on Machine Learning*, pages 3059–3068.
- Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- Salima Medhaffar, Fethi Bougares, Yannick Esteve, and Lamia Hadrach-Belguith. 2017. Sentiment analysis of tunisian dialects: Linguistic resources and experiments. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 55–61.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.
- Ahmed Mourad and Kareem Darwish. 2013. Subjectivity and sentiment analysis of modern standard arabic and arabic microblogs. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 55–64.
- Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical*

*Methods in Natural Language Processing*, pages 2515–2519.

Karim Sayadi, Marcus Liwicki, Rolf Ingold, and Marc Bui. 2016. Tunisian dialect and modern standard arabic dataset for sentiment analysis : Tunisian election context. In *To appear in the ACLing 2016 IEEE proceedings*. CICLING.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565.

Lyndon White, Roberto Togneri, Wei Liu, and Mohammed Bennamoun. 2015. How well sentence embeddings capture meaning. In *Proceedings of the 20th Australasian Document Computing Symposium*, page 9. ACM.

Mohamed A. Zahran, Ahmed Magooda, Ashraf Y. Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015. Word representations in vector space and their applications for arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 430–443. Springer.